

Solution1:

1. We know that C is a compiled language.
2. It means when it is run code will not be changing. That's why it is very speedy.
3. If we Use the Interpreter instead of compiler then the speed of the program is too slow.
4. Interpreter Translate the program line by line and program reading speed will be slow.
5. Using Interpreter for C++ compiling is very badly affected by C++ efficiency.
6. The interpreter can Increase the speed of Program execution while compilers slow

Solution2:

1. The interpreter translates one statement at a time so; its program execution is slow.
2. Interpreter overall execution time is slow that's why it is less efficient.
3. Interpreters do not generate any intermediate code, during the interpretation process. Thus, interpreters are memory efficient.
4. Interpreters stop the interpretation process if an error occurred thus, an interpreter is better to resolve the error and then to continue further execution.
5. The interpreter takes less time to analyze the source code.

6. A compiler is better for c ++ as compare to Interpreter because it uses JavaScript, Python, and other programming languages.

Solution3:

1. Using an interpreter rather than compiler to run C++, we face the problems of slow processing speed.

This is a translator reads every source code line by line, then rectifying the issues/errors are easier but it hits the goal of program optimization due to the unavailability of the overall picture of the program, ultimately making it inefficient. Execution time or running time is slower due to the behavior to go through source code line by line. Input-Interpreter takes every single line of source code as input. In the end, the interpreters are not the best option for the production environment.

Solution4:

1. The interpreter translates the program one statement at a time. So it affects the speed comparing with the compiler.
2. No intermediate object code is generated, hence are memory efficient.
3. Compiler displays all errors after compilation; on the other hand the 4. The interpreter displays errors of each line one by one.
5. The interpreter is much slower as if an error comes first we will correct it then go ahead.

Solution5:

1. Compiler code runs faster = Interpreted code run slower...!!
Runtime

2. The program code is already translated into machine code.
Thus, code execution time is less = Interpreters are easier to use, especially for beginners..!! Advantages

3. It takes the entire program = it takes a single line of code...!!
Input

4. Display all errors after compilation all at the same time. =
Displays all errors of each line one by one...!! Errors

5. The processing of compilation is relatively complicated It spends a lot of time analyzing and processing the program.. =
Relatively less time spent on analyzing and processing the program

6. It requires more memory as it generates extra object code. =
It requires less memory as it does not generate any intermediate code....!! Generate code

Key Differences between Compiler and Interpreter

Let's look at major differences between Compiler and Interpreter.

1. The compiler takes a program as a whole and translates it, but interpreter translates a program statement by statement.
2. Intermediate code or target code is generated in case of a compiler. As against interpreter doesn't create intermediate code.
3. A compiler is comparatively faster than Interpreter as the compiler take the whole program at one go whereas interpreters compile each line of code after the other.

4. The compiler requires more memory than interpreter because of the generation of object code.
5. Compiler presents all errors concurrently, and it's difficult to detect the errors in contrast interpreter display errors of each statement one by one, and it's easier to detect errors.
6. In compiler when an error occurs in the program, it stops its translation and after removing error whole program is translated again. On the contrary, when an error takes place in the interpreter, it prevents its translation and after removing the error, translation resumes.
7. In a compiler, the process requires two steps in which firstly source code is translated to target program then executed. While in Interpreter It's a one-step process in which Source code is compiled and executed at the same time.
8. The compiler is used in programming languages like C, C++, C#, Scalar, etc. On the other Interpreter is employed in languages like PHP, Ruby, Python, etc.